

## REMARKS

In response to the above identified Office Action, Applicant has amended their application and respectfully request reconsideration thereof.

Claims 102-125 have been canceled. Claims 126-142 have been added.

### Claim Rejections Under 35 U.S.C. § 103(a)

Claims 102-107, 110-115, and 118-123 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,875,332 to Wang, et al. ("Wang"). Also, claims 108-109, 116-117, and 124-125 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,875,332 to Wang, et al. ("Wang") in view of U.S. Patent No. 5,892,905 to Brandt, et al. ("Brandt"). The Applicant respectfully disagrees. Nevertheless, these claims have been canceled. The Applicant believes that new claims 126-142 are more clear and easy to understand, so that it will become more apparent that they are allowable over the prior art.

The teachings of Wang, the primary reference cited by the Examiner can be summarized as follows. Wang teaches the automatic generation of an interface between a user-generated form and a user-generated procedure. The "procedure" is the program that executes various actions on the data received from the input fields of the form. This procedure is analogous to what is referred to as the "program code" and "CGI program" in the present application.

According to Wang, it is the user that generates both the form and the procedure. Thus, Wang teaches away from generating the procedure or program code automatically. However, Wang does teach the automatic generation of the interface, known as the CGI adaptor in Wang, between the user-generated form and procedure. This is done by parsing the stored procedure to identify its arguments, and then extracting the input field names from the user-generated form and associating them with the arguments of the procedure. This is helpful, but not nearly as user-friendly as automatically generating the procedure without having the user write it, as claimed in the present invention.

In contrast, the present invention is directed to a situation where the user only composes the form, not the program code or procedure. Instead, the program code is generated automatically by the system based on user feedback through a user-friendly graphical interface that dynamically adapts to the input fields of the form according to the present invention. That is, the system of the present invention parses only the form, and generates a user interface (such as the one in Figure 4) to let the user associate certain actions with various input fields. The interface is based on the input fields found in the form. The users input from the user interface is then used to automatically generate to program code that corresponds to the user-generated procedure in Wang.

Specifically, new claim 126 requires: 1. "receiving a form authored by a user using a form authoring language, the form containing one or more input fields." In contrast, Wang also teaches the submission of a user-authored

procedure, not just the form. Claim 126 further requires: 2. "parsing the received form to identify the input fields contained in the received form." In contrast, Wang also parses the user-authored procedure to identify its arguments.

Claim 126 further requires: 3. "providing a graphical user interface to the user to allow the user to identify actions to be associated with the identified input fields, the provided user graphical user interface being dependent on the identified input fields." Wang nowhere teaches or suggests such a requirement. The input supplied by the user through this dynamic user interface will be used to generate the program code. Since in Wang the user generates the program code, this step would make no sense. Thus, Wang teaches away from the present invention. Finally, claim 126 requires: 4. "automatically generating a program code to carry out the actions identified by the user." Once again, Wang nowhere teaches or suggests this step. As mentioned above, Wang explicitly teaches away from this requirement by asking for a user-generated procedure.

Therefore, claim 126 is distinguished over Wang. The Examiner also cites Brandt, but only as a secondary reference. Brandt has been discussed in previous responses and is not claimed to teach automatic generation of program code to process form input by the Examiner. Therefore claim 126 also distinguishes over Brandt and is allowable. Claims 127-131 are dependent on allowable claim 126 adding further limitations and are thus also allowable. Independent claims 132 and 137 include limitations substantially similar to those of claim 126 and are

therefore also allowable. Claims dependent on these allowable claims are also allowable.

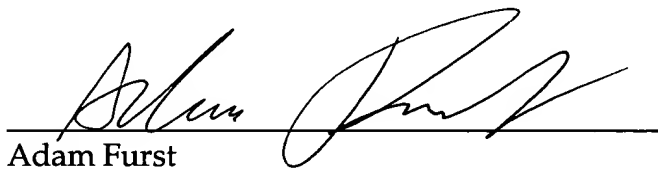
Therefore, all claims remaining in the application should now be allowed.

If the Examiner believes a telephone interview would expedite the prosecution of this application, the Examiner is invited to contact Adam Furst at (408) 947-8200 ext. 212.

If there are any additional charges, please charge them to Deposit Account No. 02-2666.

Respectfully submitted,  
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Dated: 2/7, 2003

  
Adam Furst  
Reg. No. 51,710

12400 Wilshire Blvd.  
Seventh Floor  
Los Angeles, CA 90025-1026  
(408) 947-8200

**VERSION WITH MARKINGS TO SHOW CHANGES MADE**

A marked up version of the claims is provided below.

Additions are indicated with "\_\_\_" and deletions are indicated within "[ ]."

102. (Cancelled) [A method comprising:

a first author creating a first form, the first form including a set of fields ,  
each field having one or more field attributes, the set of fields constituting a  
format specification of the first form;

a first server presenting a first user interface to a first configurer, the first  
user interface being used by the first configurer to provide the first form to the  
first server,;

the first server parsing the first form provided by the first configurer to  
extract format specification information of the first form; and

the first server presenting to the first configurer a second user interface  
which is constructed based on the format specification information extracted  
from the first form, the second user interface being used by the first configurer to  
configure a set of functions to be performed by the first server in processing  
submissions of the first form, the set of functions including one or more functions  
being responsive to the values of one or more fields contained in the submissions  
of the first form.]

103. (Cancelled) [The method of claim 102 further including:

generating a first program including one or more program components to perform the set of functions configured by the first configurer in processing a submission of the first form submitted by a first user, the first program being designated as the program to process submissions of the first form.]

104. (Cancelled) [The method of claim 102 further including:

generating a set of directives corresponding to the set of functions configured by the first configurer, the set of directives to be used by a first program that is designated as the program to process submissions of a plurality of forms including submissions of the first form.]

105. (Cancelled) [The method of claim 104 wherein the first program is configured to process submissions of multiple forms based upon multiple sets of directives, each set of directives corresponding to the set of functions configured for each form.]

106. (Cancelled) [The method of claim 102 wherein a configuration data structure is created for the first form to include configuration information provided by the first configurer with respect to the set of functions to be performed in response to a submission of the first form.]

107. (Cancelled) [The method of claim 106 further including:

maintaining consistency between the configuration data structure and the format specification of the first form.]

108. (Cancelled) [The method of claim 107 wherein maintaining consistency includes:

modifying the configuration information in the configuration data structure in response to changes in the format specification of the first form.]

109. (Cancelled) [The method of claim 107 wherein maintaining consistency includes:

determining whether the format specification of the first form has been changed since the configuration data structure was created; and

if the format specification of the first form has been changed since the configuration data structure was created, updating the configuration data structure to reflect the changes that have been made to the format specification of the first form.]

110. (Cancelled) [A system comprising:

logic to present a first user interface to a first configurer , the first user interface being used by the first configurer to provide a first form to the system, the first form being created by a first author and including a set of fields, each field having one or more field attributes, the set of fields constituting a format specification of the first form;

logic to parse the first form provided by the first configurer to extract format specification information of the first form; and

logic to present to the first configurer a second user interface which is constructed based on the format specification information extracted from the first form, the second user interface being used by the first configurer to configure a set of functions to be performed by the system in processing submissions of the first form, the set of functions including one or more functions being responsive to the values of one or more fields contained in the submissions of the first form.]

111. (Cancelled) [The system of claim 110 further including:

logic to generate a first program including one or more program components to perform the set of functions configured by the first configurer in processing a submission of the first form submitted by a first user, the first program being designated as the program to process submissions of the first form.]

112. (Cancelled) [The system of claim 110 further including:

logic to generate a set of directives corresponding to the set of functions configured by the first configurer, the set of directives to be used by a first program that is designated as the program to process submissions of a plurality of forms including submissions of the first form.]

113. (Cancelled) [The system of claim 112 wherein the first program is configured to process submissions of multiple forms based upon multiple sets of directives, each set of directives corresponding to the set of functions configured for each form.]



114. (Cancelled) [The system of claim 110 wherein a configuration data structure is created for the first form to include configuration information provided by the first configurer with respect to the set of functions to be performed in response to a submission of the first form.]

115. (Cancelled) [The system of claim 114 further including:

logic to maintain consistency between the configuration data structure and the format specification of the first form.]

116. (Cancelled) [The system of claim 115 wherein logic to maintain consistency includes:

logic to modify the configuration information in the configuration data structure in response to changes in the format specification of the first form.]

117. (Cancelled) [The system of claim 115 wherein logic to maintain consistency includes:

logic to determine whether the format specification of the first form has been changed since the configuration data structure was created; and

logic to update the configuration data structure to reflect changes that have been made to the format specification of the first form if the format specification of the first form has been changed since the configuration data structure was created.]

118. (Cancelled) [A machine-readable medium comprising instructions which, when executed by a machine, cause the machine to perform operations including:

a first server presenting a first user interface to a first configurer , the first user interface being used by the first configurer to provide a first form to the first server, the first form being created by a first author and including a set of fields , each field having one or more field attributes, the set of fields constituting a format specification of the first form;

the first server parsing the first form provided by the first configurer to extract format specification information of the first form; and

the first server presenting to the first configurer a second user interface which is constructed based on the format specification information extracted from the first form, the second user interface being used by the first configurer to configure a set of functions to be performed by the first server in processing submissions of the first form, the set of functions including one or more functions being responsive to the values of one or more fields contained in the submissions of the first form.]

119. (Cancelled) [The machine-readable medium of claim 118 further including:

generating a first program including one or more program components to perform the set of functions configured by the first configurer in processing a submission of the first form submitted by a first user, the first program being designated as the program to process submissions of the first form.]

120. (Cancelled) [The machine-readable medium of claim 118 further including:  
generating a set of directives corresponding to the set of functions  
configured by the first configurer, the set of directives to be used by a first  
program that is designated as the program to process submissions of a plurality  
of forms including submissions of the first form.]

121. (Cancelled) [The machine-readable medium of claim 120 wherein the first  
program is configured to process submissions of multiple forms based upon  
multiple sets of directives, each set of directives corresponding to the set of  
functions configured for each form.]

122. (Cancelled) [The machine-readable medium of claim 118 wherein a  
configuration data structure is created for the first form to include configuration  
information provided by the first configurer with respect to the set of functions  
to be performed in response to a submission of the first form.]

123. (Cancelled) [The machine-readable medium of claim 122 further including:  
maintaining consistency between the configuration data structure and the  
format specification of the first form.]

124. (Cancelled) [The machine-readable medium of claim 123 wherein  
maintaining consistency includes:  
modifying the configuration information in the configuration data  
structure in response to changes in the format specification of the first form.]

125. (Cancelled) [The machine-readable medium of claim 123 wherein maintaining consistency includes:

determining whether the format specification of the first form has been changed since the configuration data structure was created; and

if the format specification of the first form has been changed since the configuration data structure was created, updating the configuration data structure to reflect the changes that have been made to the format specification of the first form.]

126. (New) A method comprising:

receiving a form authored by a user using a form authoring language, the form containing one or more input fields;

parsing the received form to identify the input fields contained in the received form;

providing a graphical user interface to the user to allow the user to identify actions to be associated with the identified input fields, the provided user graphical user interface being dependent on the identified input fields;

automatically generating a program code to carry out the actions identified by the user.

127. (New) The method of claim 126, wherein the form authoring language is Hyper Text Markup Language (HTML).

128. (New) The method of claim 126, wherein the generated program code is a Common Gateway Interface (CGI) program.

129. (New) The method of claim 126, further comprising associating the generated program code with the form.

130. (New) The method of claim 129, further comprising:

receiving a specific instance of the form, including data from the input fields; and

executing the program code on the received data.

131. (New) The method of claim 126, further comprising periodically determining whether the generated program code is consistent with the form; and

generating an alert if the generated program code is not consistent with the form.

132. (New) A server comprising:

a communications device connected to a network to receive a form from a client connected to the network, the form authored by a user using a form authoring language and containing one or more input fields;

a memory coupled to the communications device to store the received form;

a parser module coupled to the memory to parse the received form to identify the input fields contained in the received form;

a configurer module coupled to the parser module to create a graphical user interface based on the input fields identified by the parser module, and to provide the graphical user interface to the user using the communications device to allow the user to identify actions to be associated with the identified input fields;

a code generator module coupled to the configurer module to automatically generate a program code to carry out the actions identified by the user.

133. (New) The server of claim 132, further comprising a processor, wherein the parser module, the configurer module, and the code generator module are all implemented using the processor.

134. (New) The server of claim 132, wherein the form authoring language is Hyper Text Markup Language (HTML).

135. (New) The server of claim 132, wherein the generated program code is a Common Gateway Interface (CGI) program.

136. (New) The server of claim 132, further comprising a consistency module to periodically determine whether the generated program code is consistent with the form, wherein the consistency module generates an alert if the program code is not consistent with the form.

137. (New) A machine-readable medium containing data representing instructions that, when executed by a processor, cause the processor to perform operations comprising:

receiving a form authored by a user using a form authoring language, the form containing one or more input fields;

parsing the received form to identify the input fields contained in the received form;

providing a graphical user interface to the user to allow the user to identify actions to be associated with the identified input fields, the provided user graphical user interface being dependent on the identified input fields;

automatically generating a program code to carry out the actions identified by the user.

138. (New) The machine-readable medium of claim 137, wherein the form authoring language is Hyper Text Markup Language (HTML).

139. (New) The machine-readable medium of claim 137, wherein the generated program code is a Common Gateway Interface (CGI) program.

140. (New) The machine-readable medium of claim 137, wherein the instructions further cause the processor to perform operations comprising associating the generated program code with the form.

141. (New) The machine-readable medium of claim 140, wherein the instructions further cause the processor to perform operations comprising:

receiving a specific instance of the form, including data from the input fields; and

executing the program code on the received data.

142. (New) The machine-readable medium of claim 137, wherein the instructions further cause the processor perform operations comprising periodically determining whether the generated program code is consistent with the form; and

generating an alert if the generated program code is not consistent with the form.